浮动

1. 浮动的简介

- 通过浮动可以使一个元素向其父元素的左侧或右侧移动
- 使用float属性来设置于元素的浮动
- none 默认值，元素不浮动
- left 元素向左浮动
- right 元素向右浮动

注意：

- 元素设置浮动以后，水平布局的等式便不需要强制成立
- 元素设置浮动以后，会完全从文档流中脱离，不再占用文档流的位置，所以元素下边的还在文档流中的元素会自动向上移动

2. 浮动的特点

- 浮动元素会完全脱离文档流，不再占据文档流中的位置
- 设置浮动以后，元素会向父元素的左侧或右侧移动
- 浮动元素默认不会从父元素中移出
- 浮动元素向左或向右移动时，不会超过前边的浮动元素（先来后到的顺序）
- 浮动元素不会超过上边的浮动的兄弟元素，最多就是和它一样高
- 如果浮动元素的上边是一个没有浮动的块元素，则浮动元素无法上移
- 浮动元素不会盖住文字，文字会自动环绕在浮动元素的周围，所以我们可以利用浮动来设置文字环绕图片的效果

3. 浮动存在的问题

1. 高度塌陷：

- 在浮动布局中，父元素的高度默认是被子元素撑开的
- 当子元素浮动后，其会完全脱离文档流，子元素从文档流中脱离将会无法撑起父元素的高度，导致父元素的高度丢失， 父元素高度丢失以后，其下的元素会自动上移，导致页面的布局混乱

2. 解决方法：

- 给父元素一个高度
- 在浮动元素后面给一个空标签，类名叫clear

```
.clear {
clear: both;
}
```

- 给父元素增加伪元素选择器::after

高度塌陷问题，一般用::after

外边距重叠问题，一般用::before

```
.clearfix::before,
.clearfix::after{
    content: '';
    display: table;
    clear: both;
}
```

- 给父元素增加overflow:hidden

4. 背景

1. background-color 设置背景颜色
2. background-image 设置背景图片

- 如果背景图片大小小于元素，则背景图片会自动在元素中平铺将元素铺满
- 如果背景图片大小大于元素，则背景图片一部分会无法完全显示
- 如果背景图片大小等于元素，则背景图片会直接正常显示

3. background-repeat 设置背景图片的重复方式

- repeat 默认值，背景图片沿着x轴和y轴双方向平铺
- repeat-x 背景图片沿着x轴方向平铺
- repeat-y 背景图片沿着y轴方向平铺
- no-repeat 背景图片不平铺

4. background-position 设置背景图片的位置

- 通过top left right bottom center几个表示方位的词来设置背景图片的位置：使用方位词时必须要同时指定两个值，如果只写一个则第二个默认就是center
- 通过偏移量来指定背景图片的位置：水平方向偏移量、垂直方向变量

5. background-clip 设置背景的范围

- border-box 默认值，背景会出现在边框的下边
- padding-box 背景不会出现在边框，只出现在内容区和内边距
- content-box 背景只会出现在内容区
- background-origin 背景图片的偏移量计算的原点
- border-box 背景图片的变量从边框处开始计算
- padding-box 默认值，background-position从内边距处开始计算
- content-box 背景图片的偏移量从内容区处计算

6. background-size 设置背景图片的大小

- 第一个值表示宽度，第二个值表示高度；如果只写一个，则第二个值默认是auto
- cover 图片的长宽比例放大，将元素铺满
- contain 图片长宽比例放大，将图片在元素中完整显示
- 百分比 100% 100% 图片完全填满并显示在盒子里面

7. background-attachment 背景图片是否跟随元素移动

- scroll 默认值，背景图片会跟随元素移动
- fixed 背景会固定在页面中（浏览器某个位置），不会随元素移动

8. 多背景缩写

- 图片地址可以用逗号隔开
- 注意：背景缩写时， 如果要使用background-size一定要写在position的后面，并且+/

```
background: url(./img/app,png) fixed  no-repeat  0px 0px  / cover,url();
```

精灵图

1. 解决图片闪烁的问题：

- 可以将多个小图片统一保存到一个大图片中，然后通过调整background-position来显示响应的图片
- 这样图片会同时加载到网页中就可以有效的避免出现闪烁的问题
- 这个技术在网页中应用十分广泛，被称为CSS-Sprite，这种图我们称为雪碧图

2. **精灵图**的使用步骤：

- 先确定要使用的图标
- 测量图标的大小
- 根据测量结果创建一个元素
- 将**精灵图**设置为元素的背景图片
- 设置一个偏移量以显示正确的图片

3. **精灵图**的特点：

- 一次性将多个图片加载进页面，降低请求的次数，加快访问速度，提升用户的体验

定位

1. 定位 position

- static 静态定位(默认) 一般用来清除定位
- relative 相对定位
- absoluted 绝对定位
- fixed 固定定位
- sticky 粘滞定位

1. 偏移量

top bottom left right

- top值越大，定位元素越靠下
- bottom值越大，定位元素靠上
- left越大，定位元素越靠右
- right越大，定位元素越靠左
- 单位：px

- 百分比 50% 50%

定位要配合偏移量来使用

2. 相对定位的特点

- 当元素开启相对定位以后，如果不设置偏移量元素，则元素不会发生任何变化，移动后，原来的位置还在
- 相对定位是参照于元素在文档流中的位置进行定位的
- 相对定位会提升元素的层级
- 相对定位不会改变元素的性质：块还是块，行内还是行内

3. 绝对定位的特点

- 开启绝对定位后，如果不设置偏移量，元素的位置不会发生变化，原来的位置不在了
- 开启绝对定位后，元素会从文档流中脱离
- 绝对定位会改变元素的性质：行内变成块，块的宽高被内容撑开
- 绝对定位会使元素提升一个层级
- 绝对定位元素是相对于其父级或者祖先元素非static定位进行定位，如果父级或祖先没有定位就相对于浏览器进行定位

4. 固定定位的特点

- 固定定位也是一种特殊的绝对定位，脱离文档流，原来的位置不在了
- 固定定位参照于浏览器进行定位，不会随网页的滚动条滚动

2. z-index 层级

- 层级z-index只针对同级别的，不会影响嵌套的层级。嵌套的最外层在最下面的层级。同级别的可以通过z-index来改变它们之间的层级，同级别要使用z-index,必须要用relative,absolute,fixed定位，然后z-index才能生效。
- 给父级设置层级会影响子代的层级
- 给子代设置层级不会影响父级的层级

3. 绝对定位水平居中

- left50% 走父容器宽度的一半
- margin 负值 往左边走自己盒子宽度的一半

```
<style>
  .box {
    width: 200px;
    height: 200px;
    background-color: pink;
    /* margin: auto; */

    position: absolute;
    /* 1.left走50% 父容器宽度的一半 */
    left: 50%;
```

```
    /* 2.margin 负值 往左边走 自己盒子宽度的一半 */
    margin-left: -100px;

    top: 50%;
    margin-top: -100px;

 }
    </style>
</head>

<body>
    <div class="box"></div>
</body>
```

4. 显示与隐藏

- 父盒子设置相对定位，子盒子设置绝对定位；子盒子用 display: none 隐藏，在使用伪类选择器 :hover 和 display: block 显示盒子

```
<style>
  * {
     margin: 0;
     padding: 0;
 }

  .fa {
      position: relative;
      width: 200px;
      height: 200px;
      background-color: red;
  }

  .son {
      display: none;
      position: absolute;
      top: 220px;
      width: 200px;
      height: 200px;
      background-color: green;
```

```
    }

    .fa:hover .son {
        display: block;
    }
    </style>
</head>
<body>
    <div class="fa">
        <div class="son"></div>
    </div>
</body>
```

5. 盒子阴影

1. 第一个值——水平偏移量：设置阴影的水平位置
- 正值向右移动
- 负值向左移动
2. 第二个值——垂直偏移量：设置阴影的垂直位置
- 正值向下移动
- 负值向上移动
3. 第三个值——阴影的模糊半径
4. 第四个值——阴影的颜色

```
    <style>
        * {
            margin: 0;
            padding: 0;
        }

        .demo {
            position: absolute;
            top: 50%;
            left: 50%;
            height: 200px;
            width: 200px;
            background-color: red;
            /* 盒子阴影 */
            box-shadow: 0px 0px 10px #999;
```

```
16
17          }
18
19      </style>
20  </head>
21  <body>
22      <div class="demo"></div>
23  </body>
```